Faculty Mission:
Enable 49 ML & AI Solutions
That 1.4 Billion People Are Waiting For

Audience:

BE & ME Faculty | EEE | ECE | CSE |
ML/AI Departments

The One Change Faculty Engineer Can Make

Require every student to build ONE complete, working, real-world product – from concept to tested product – Not in one semester – Not in one year – Make use of all the 4 Years

What the Product Must Include

- Problem definition & design calcs
- Circuit design / mechanical design
- PCB / CAD creation
- Prototype fabrication & debugging
- Testing with measured data
- BOM cost estimation
- Compliance pre-checks (EMI, safety)

Why This Differs from Typical Projects

- No purchased kits – original design required
- Must physically work and be measured
- Failure analysis is mandatory
- Iterative improvements documented
- Cross-disciplinary teams encouraged
- Faculty grade the working demo, not just report

Suggested Evaluation Weighting

- 20% – Concept & design review
- 20% – Prototype implementation
- 20% – Debugging & improvements
- 20% – Testing & measured results
- 10% – Documentation
- 10% – Final working demonstration

Example Projects by Branch

- ECE/EEE: DC-DC converters, motor controllers, BMS, Embedded Systems that are AI ready
- CSE: Deployable IoT software with tests & monitoring, Deploy ML on Embedded systems
- Mechanical: Gearbox / actuator with load testing
- Civil: Structural health sensor or water filter model
- All: IoT sensor node with AI and cloud dashboard

I believe faculties embracing this change will be felt & respected by Students & Nation, rather than leaving it for natural/forceful correction

Faculty Advantage: Generative AI as Your Industry Co-Mentor

You don't need 20 years of industry experience to guide students on EMC, environmental testing, or product certification — you need the right AI tools and the confidence to use them alongside your students. ***It is also essential to allocate time to reach out to hands-on proven professionals, may be your alumni***

The Honest Reality: Where Faculty Stand Today

<15%

Engineering faculty regularly use GenAI tools in teaching

>70%

Students use ChatGPT / Gemini / Copilot / Claude / Grok weekly

3–5 yrs

Average gap between industry practice & college syllabus

Today

Best time to close the gap — tools are free or low cost

Four Ways Faculty Can Use GenAI Right Now — Starting This Week

EMC & Signal Integrity Guidance

- Prompt: 'My student's PCB has a 48MHz MCU — list the top 5 EMC risks and how to mitigate them in layout'
- Ask for decoupling capacitor selection, trace impedance, and return path analysis
- Get IEC 61000-4 test explanations with plain-language pass/fail criteria
- Generate pre-compliance checklists tailored to the specific circuit
- Tools: Claude, ChatGPT-4o, Gemini Advanced

Environmental & IP Rating Design

- Ask for IP54/IP65/IP67 enclosure design requirements for a given application
- Get material selection guidance for tropical, humid or dusty Indian conditions
- Generate thermal calculations for enclosure sizing
- Get IEC 60068 test standards (temp cycling, humidity, vibration) explained simply
- Tools: Claude, ChatGPT-4o

Component Selection & BOM Review

- Upload a student's schematic or BOM — ask AI to flag design risks
- Get alternate component suggestions for obsolete or out-of-stock parts
- Ask for worst-case tolerance analysis on regulators or filter circuits
- Identify single points of failure in a power supply design
- Tools: Claude (with file upload), ChatGPT-4o

Certification & Compliance Roadmap

- Ask: 'What certifications does a soil moisture IoT device sold in India need?'
- Get a step-by-step BIS / WPC / CE / FCC pre-compliance checklist
- Generate a test plan for a student product targeting a specific standard
- Understand cost and timeline for each certification stage
- Tools: Claude, Perplexity (for live regulatory search)

Sample Prompts to Try Today

- *"Explain conducted EMI and how a student can reduce it in a SMPS project — use simple language"*
- *"Review this BOM for a 12V motor controller — flag any missing protection components"*
- *"What environmental tests must a water-quality sensor pass for outdoor use in India?"*
- *"Generate a 10-point design review checklist for a student building a LoRaWAN sensor node"*

3 Actions This Week

1. Pick one project — Open Claude or ChatGPT — ask it to review a student schematic for EMC risks.
2. Run a mock design review — Ask AI for a 10-point checklist. Grade the student's design against it.
3. Build a prompt library — Save one useful AI response per week. Share it across the department.

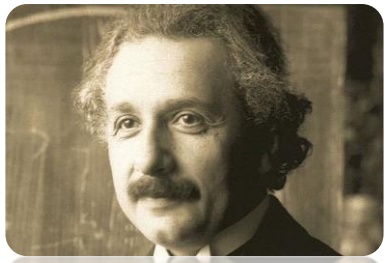
The student who builds an EMC-aware design with AI assistance learns more than one who copies a kit. The faculty who guides that process becomes an industry-relevant mentor — without needing a full fledged factory.

Faculty - The Sovereign Change Agent

Faculty, the eagle,
can fly above all the barriers to “Reclaim the Joy of Engineering”

“Many faculties may be overwhelmed by responsibilities already.
That is exactly why this proposal does not add a single line of paperwork.
We are not changing the syllabus immediately; we are changing the execution.
Watch that change in our students as they turn as “fellow engineers” journeying with us for 4 years.
Not just Student, the Faculty will enjoy the thrill of building things that works flawless.

"The purpose of education is to create good human beings with skill and expertise. Enlightened human beings can be created by teachers." – **Dr. APJ Abdul Kalam**



"The value of a college education is not the learning of many facts, but the training of the mind to think." – **Albert Einstein**

India Engineering Product Platform Initiative

IEPPI

A Four-Year Faculty-Led Framework
for Hands-On Engineering Education in India

Mission

Transforming India's 3,500 Engineering Colleges into Product-Building Ecosystems

~15L B.Tech seats/year	**~3,500** Engineering colleges
Only ~5L Core engineering jobs	**5 Years** Transform the gap

Presented to Engineering Faculty Leaders across India

IEPPI - Platform Architecture

INTELLIGENT CONNECTED SYSTEMS

Analogous to Intelligent Humans and Many Humans Collaborating in a Society

An Intelligent Connected System senses, thinks, acts and communicates. Many such systems connect and collaborate to create collective intelligence.

1. THE INDIVIDUAL – INTELLIGENT HUMAN

A single human is intelligent and can act independently.

SENSES
(Eyes, Ears, Skin, etc.)
Perceive the environment

MIND
(Brain)
Thinks, learns, decides and plans

ACTIONS
(Muscles & Limbs)
Executes actions in the real world

ENERGY
(Heart)
Provides and sustains energy

A human can sense, think, act and survive on their own.

2. THE INTELLIGENT SYSTEM – THE INDIVIDUAL AGENT

An intelligent system can operate autonomously.

SENSORS
Capture data from the environment

EDGE AI
(MCU/MPU/NPU + AI/ML)
Processes data locally, learns, decides and makes predictions

ACTUATORS
Executes actions through motors, valves, relays, etc.

POWER SYSTEM
Supplies and manages energy

Autonomous · Adaptive · Real-time · Reliable

An intelligent system senses, thinks, acts and operates independently.

3. THE COMMUNICATION CHANNEL

Enables systems to share information.

CONNECTIVITY
(Wired, Wireless, LPWAN, Cellular, Bluetooth, etc.)
Transmits data to and from other systems

PROTOCOLS
(Language & Rules)
Defines how systems communicate and understand each other

GATEWAYS
Bridges between local systems and remote networks / cloud

SECURITY
Ensures authentic, secure and reliable communication

The communication channel enables exchange of information and collaboration.

4. THE SOCIETY – COLLECTIVE INTELLIGENCE

Many intelligent systems collaborate to solve bigger problems and create greater impact.

Multiple systems connect, share, learn and make better decisions – together as a society.

STANDALONE vs CONNECTED

STANDALONE INTELLIGENT SYSTEM
(Like a single individual)

- Operates independently
- Makes local decisions
- Works without connectivity
- Suitable for real-time and offline use

CONNECTED INTELLIGENT SYSTEMS
(Like people in a community)

- Share data and context
- Collaborate and coordinate
- Learn from collective data
- Create system-level insights

EXAMPLE APPLICATIONS

Smart Irrigation · Predictive Maintenance · Patient Monitoring · Traffic Management
Fleet Tracking · Smart Buildings · Environmental Monitoring · Energy Management

MAPPING TABLE: HUMAN ANALOGY vs INTELLIGENT CONNECTED SYSTEMS

Human Analogy	Intelligent Connected Systems	Functional Role	Required for Standalone Operation?
Senses (Eyes, Ears, Skin)	Sensors	Observe the environment	Yes
Mind / Intelligence	Edge AI (MCU/MPU/NPU + AI/ML)	Think, learn, decide	Yes
Actions (Muscles & Limbs)	Actuators	Execute actions	Yes / Optional
Energy (Heart)	Power System	Supply and sustain energy	Yes
Communication (Voice, Language)	Connectivity (Wired / Wireless)	Share information	No
Society / Community	Multiple Systems & Cloud	Collective intelligence	No

KEY TAKEAWAY

An Intelligent Connected System is like an intelligent human – it can sense, think, act and survive on its own. Connectivity enables many such systems to collaborate and create collective intelligence – like a society.

THE COMPLETE ECOSYSTEM

Individual Intelligent System (Autonomous) + Communication Channel (Share & Exchange) + Collective Intelligence (Analyze & Learn) = Smarter Society · Better Decisions · Greater Impact

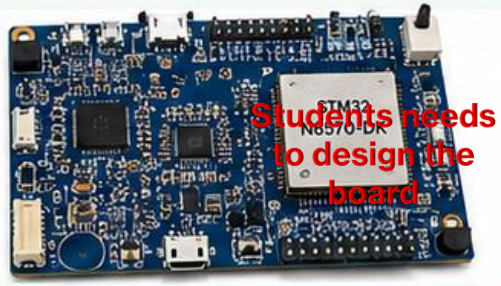
IN ONE LINE

From intelligent individuals to a connected society – Intelligent Connected Systems make the world smarter, safer and more sustainable.

The Embedded AI Platform Family — EAP-1 to EAP-3

EAP-1: TinyML Node

STM32N6 · Ethos-U55 NPU · 4.8 TOPS · No Linux · Battery powered · Real-time



- Ultra low power
- Real-time Inference
- Wireless Connectivity

Design element	Specification
Processor	STM32N6 — Cortex-M55 + Ethos-U55 NPU
AI framework	X-CUBE-AI — Keras/TF/ONNX → optimised C
Best for	Audio classification · vibration anomaly · basic image
Power	µA sleep · mA inference · years on battery
Connectivity	LoRa STM32WL or BLE nRF5340
POC board	STM32N6570-DK (₹6,000–9,000)
Engineering target	Custom 4-layer PCB + STM32N6 chip

Solutions

- MF-1 (MSME vibration)
- GR-3 (pest acoustic)
- AG-4 (livestock)
- TR-3 (railway AE)

EAP-2: Edge AI Gateway

STM32MP2 / RPi CM4 · Linux + real-time · Multi-sensor fusion · Government API

- Cloud / API Ready
- Multi-sensor Fusion
- Secure & Updatable

Design element	Specification
Processor	STM32MP257 (A35+M33+NPU) or RPi CM4
AI framework	ONNX Runtime · TF Lite · Python on Linux
Best for	Multi-sensor fusion · government API · MQTT broker · OTA
Power	Mains or large battery + solar
Connectivity	NB-IoT · LoRa gateway · 4G · WiFi
POC board	STM32MP157F-DK2 or RPi CM4 + IO Board
Engineering target	Custom carrier board — see Appendix C-14

Solutions

AG-1 · HC-2 · SC-1 · EN-1 · WE-1 · GV-1

— most of A-series

EAP-3: Vision AI Node

Jetson Orin Nano · 40 TOPS · CUDA · Multi-camera · DeepStream

- Multi-camera Vision
- High Performance Inference
- GPU Acceleration

Design element	Specification
Processor	NVIDIA Jetson Orin Nano (40 TOPS)
AI framework	NVIDIA DeepStream · YOLO · TensorRT
Best for	Defect detection · ANPR · wildlife ID · face recognition
Power	5–10W — mains or large solar
Connectivity	GigE Vision (camera) · 4G/5G · WiFi
POC board	Jetson Orin Nano Dev Kit (₹22,000–28,000)
Engineering target	Custom carrier board — thermal + camera interface

Solutions

- MF-3 (quality inspection)
- SC-6 (ANPR)
- GV-2 (border)
- WE-5 (wildlife)
- TR-5 (fatigue)

Common across all EAPs:

- NABL / BIS / ISO compliant hardware design
- Secure boot & encrypted firmware update
- Edge-to-Cloud integration
- Modular, scalable & field upgradable

IEPPI - Platform Architecture

A modular, application-agnostic hardware platform *shared across 3,500 colleges — designed from Day 1 for EMC, environmental robustness and certification pathways.*

Layer 1
Sensor Layer

Agriculture · Water · Industrial · Cold Chain

- Soil: moisture, EC, pH, NPK, leaf wetness
- Water: flow, pressure, level, turbidity, DO, conductivity
- Industrial: temperature, vibration, current, gas, RPM
- Cold Chain: temp, humidity, door state, compressor current
- Analog front-ends with calibration & traceability

Layer 2
Embedded Processing Layer

Class A (Basic) · Class B (Connected) · Class C (Edge AI)

- **Class A** – MCU, local display, data log, TinyML (STM32G4, MSPM0)
- **Class B** – Wi-Fi, BLE, LoRaWAN, NB-IoT, RS-485, CAN, OTA (ESP32, STM32H5)
- **Class C** – Linux + MCU, vision AI, edge analytics (TI AM62A, i.MX8M, Jetson Orin Nano)
- All classes: TinyML ready, secure boot, OTA update
- Open firmware stack: FreeRTOS / Zephyr, TFLite Micro, ONNX

Layer 3
Load Control Layer

Actuation · Power · Drives

- Relay & SSR outputs with isolation
- Triac dimmers & soft starters
- VFD interfaces (RS-485 / CAN)
- Brushed / BLDC motor drivers
- Solenoid & valve controllers
- H-bridge modules

A modular, application-agnostic hardware platform ; not complex; no immediate big investments for lab; Same platform but variety of solutions for India specific needs across multiple domains

How to Choose the Right AI Approach?

A Decision Tree for AI / ML / DL

Can I model it with physics/logic equations?

Yes

Pure AI
(Kalman Filter, Fuzzy Logic)
Use algorithms based on rules, equations and expert knowledge.

No, it's too complex. I need data.

Is the data structured / tabular?

Yes

ML without DL
(Regression, Random Forest)
Use traditional machine learning algorithms on structured data.

No

Is the data unstructured (Vision/Audio)?

Yes

DL + ML inside AI
(Neural Networks)
Use deep learning models that learn patterns from unstructured data.

Key Takeaway: Match the problem and data type to the right AI approach for better results, efficiency and interpretability.

Machine Learning based AI Implementation Process — End to End

1 Instrument the System

Collect structured sensor data over time.

2 Label the Data

Label the data — Normal / Fault / Degraded or continuous measurement targets.

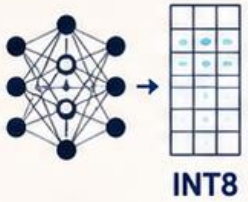
3 Select Algorithm

- Decision Tree
- Random Forest
- SVM
- Shallow Neural Network

4 Train on Workstation or Cloud

Train the model and validate accuracy against held-out data.

5 Quantize Model

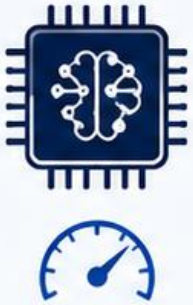


Compress the trained model to INT8 for efficient embedded deployment.

6 Deploy to MCU

Deploy via OTA, JTAG, DFU, or any supported update mechanism.

7 Inference Engine Runs on MCU



Decision made in microseconds on every new reading.

8 Retrain Periodically

As new field data is collected...

9 Retrain and Redeploy Improved Model

Use new field data to retrain, re-validate and deploy improved model continuously.

- Adaptive Improvement
- Higher Accuracy over Time
- Efficient Embedded Performance
- Lower Bandwidth & Compute Cost
- Secure & Reliable Deployment

ML based AI Implementation for Rotating Machinery Health Monitoring System

End-to-End Implementation Flow (Using STM32 MCUs & Ecosystem – Without NanoEdge AI Studio)

1 Define Problem & System

Goal: Predict machine condition in real time

Outputs (example):
- Normal
- Misalignment
- Bearing Fault
- Imbalance
- Lubrication Issue
- Degradation
- Critical Fault

2 Instrument the System

3x RTD Temperature Channels (Front Bearing, Back Bearing, Machine Body)

1x Analog Accelerometer (Vibration)

1x Acoustic Sensor (Structure-borne Sound)

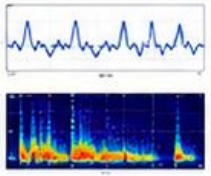
Signal Conditioning & Anti-alias Filtering

3 Data Acquisition & Logging

- STM32 ADC + DMA
- High speed sampling for vibration & acoustic
- Timestamp, RPM/Load
- Store raw data (binary/CSV)

Build a diverse dataset across operating conditions & faults

4 Data Preprocessing & Feature Engineering



- Filter, Resample
- Windowing (e.g., 1s with 50% overlap)
- Extract Features
  - Temperature: mean, std, gradients, ΔT
  - Vibration: RMS, Peak, Kurtosis, Band Energy, Envelope, etc.
  - Acoustic: FFT bins, Spectral features, MFCCs, etc.

Create feature vector for each window

5 Labeling (Ground Truth)

- Normal
- Misalignment
- Bearing Fault
- Imbalance
- Lubrication Issue
- Degradation
- Critical Fault

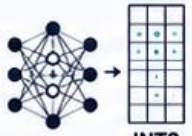
Label each window with machine condition

6 Model Training (Offline)

- Use Python (scikit-learn / TF)
- Algorithms:
  - Random Forest
  - SVM
  - Shallow Neural Network (MLP)
- Evaluate with accuracy, precision, recall, F1-score, confusion matrix

Select best performing model

7 Model Optimization & Quantization



- Convert model to TensorFlow Lite (.tflite)
- Quantize to INT8 for embedded deployment
- Target small model size & low memory footprint

Ensure accuracy retention after quantization

8 Model Conversion with STM32Cube.AI

- Import .tflite model
- Optimize for target STM32 MCU
- Generate C inference code & header files
- Get memory & performance report
- Validate model output

Ready-to-use embedded C code for inference

9 Application Development on STM32

Using STM32CubeMX & STM32CubeIDE

- **Configure peripherals** (ADC, DMA, I2C/SPI, Timers, UART, etc.)
- **Integrate X-CUBE-AI middleware**
- **Integrate CMSIS-DSP** for signal processing
- **Build application firmware**

Compile & Flash to STM32 MCU

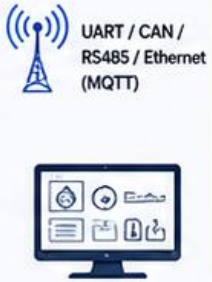
10 Real-time Inference & Decision Logic

On STM32 MCU (in production)

- Acquire sensor data
- Preprocess & extract features
- Normalize features
- Run ML inference
- Apply decision logic with thresholds & persistence
- Generate health status & alarms

Decision in microseconds on every window

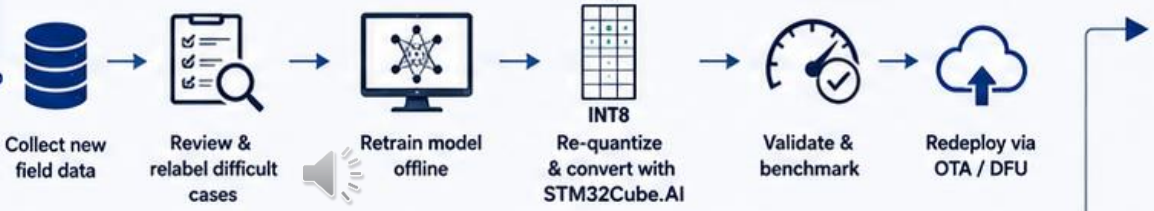
11 Deployment, Monitoring & Field Operation



- Send health status & alarms
- Visualize on dashboard (STM32CubeMonitor or custom HMI)
- Log events
- OTA / DFU updates for firmware & model

Continuous monitoring in the field

12 Continuous Improvement Loop



Benefits

- ✓ Early fault detection & reduced downtime
- ✓ Lower maintenance cost
- ✓ Improved safety & reliability
- ✓ Runs fully on STM32 MCU (edge intelligence)
- ✓ Scalable & continuously improving

Key ST Tools Used

- STM32CubeMX – Configuration & Code Generation
- STM32CubeIDE – Development Environment
- STM32Cube.AI – Model Conversion & Optimization
- STM32CubeMonitor – Monitoring & Visualization
- CMSIS-DSP – Signal Processing Library

ⓘ This flow enables an end-to-end ML based AI solution for rotating machinery health monitoring using STM32 MCUs and the ST ecosystem.

ML based AI Implementation for Rotating Machinery Health Monitoring System

End-to-End Implementation Flow (Using NanoEdge AI Studio)

- **Faster Development** with no/low-code ML
- **Optimized for STM32** from edge to cloud
- **Small Footprint** & Low Power
- **Built-in Anomaly Detection** & Classification
- **Easy Update** & Redeploy

Use Case Sensors

- 3x RTD Temperature (Front, Back Bearing, Body)
- 1x Analog Accelerometer (Vibration)
- 1x Acoustic Sensor (Structure-borne Sound)

1 Define Problem & System

Goal: Predict machine condition in real time

Outputs (example):

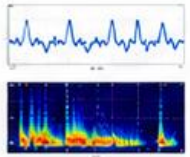
- Normal
- Misalignment
- Bearing Fault
- Imbalance
- Lubrication Issue
- Degradation
- Critical Fault

2 Instrument the System & Acquire Data

- 3x RTD Temperature Channels
- 1x Analog Accelerometer (Vibration)
- 1x Acoustic Sensor (Structure-borne Sound)

- STM32 ADC + DMA
- High speed sampling
- Timestamp, RPM/Load
- Store raw data (binary/CSV)

3 Prepare Data for Training



- Filter, Resample, Denoise
- Windowing (e.g., 1s with 50% overlap)
- Extract Features
  - Time, Frequency, Time-Frequency
  - Temperature: mean, std, gradients, ΔT
  - Vibration: RMS, Peak, Kurtosis, Band Energy, etc.
  - Acoustic: FFT bins, Spectral features, MFCCs, etc.

4 Label Data (Ground Truth)

- Normal
- Misalignment
- Bearing Fault
- Imbalance
- Lubrication Issue
- Degradation
- Critical Fault

Label each window with machine condition

5 Train with NanoEdge AI Studio

NanoEdge AI Studio

- Import dataset (features or signals)
- Select task:
  - Classification or Anomaly Detection
- Configure training
- Train & validate
- Auto model search (optional)

High accuracy with minimal effort

6 Model Evaluation & Selection

- Evaluate with built-in metrics
- Confusion matrix, F1-score, recall, precision
- Select best model based on accuracy & footprint

Choose model that meets performance target

7 Optimize for STM32

- Automatic optimization for target MCU
- Quantization (INT8)
- Pruning
- Memory & latency estimation

Small, fast and power-efficient model

8 Generate Deployment Package

- Generate optimized C code library
- Includes model weights, inference engine & APIs
- Model + metadata + performance report

Ready-to-use embedded package

9 Application Development on STM32

STM32 MCU

- Use STM32CubeMX & STM32CubeIDE
- Configure peripherals (ADC, DMA, I2C/SPI, Timers, UART, etc.)
- Integrate NanoEdge AI library
- Implement data pipeline & inference

Build firmware with NanoEdge AI library

10 Real-time Inference & Decision Logic

- Run inference on STM32 MCU
- Decision logic with thresholds & persistence
- Generate health status & alarms

Decision in microseconds on every window

11 Deployment, Monitoring & Field Operation

- Send health status & alarms
- Visualize on dashboard (STM32CubeMonitor or custom HMI)
- Log events
- OTA / DFU updates for firmware & model

Continuous monitoring in the field

12 Continuous Improvement Loop

Collect new field data → Review & relabel difficult cases → Retrain & improve model in NanoEdge AI Studio → Re-optimize for target STM32 → Redeploy via OTA / DFU

- ✓ Continually adapts to new operating conditions
- ✓ Improves accuracy over time
- ✓ Lower downtime & maintenance cost
- ✓ Higher reliability & safety

Key ST Ecosystem Tools

- STM32CubeMX
- STM32CubeIDE
- STM32CubeMonitor
- Monitoring & Visualization
- X-CUBE-AI Deployment Middleware
- CMSIS-DSP Signal Processing Library
- STM32 MCUs Optimized for Edge AI

ⓘ This flow enables an end-to-end ML based AI solution for rotating machinery health monitoring using STM32 MCUs and NanoEdge AI Studio.

Can or should a B.E. student of India work on all four layers?

Layer	Can a B.E. student work on it?	What's realistic in 3 years?	Faculty level	Student level
4 Application Dashboards, Apps, ML Models	Yes	• Build dashboards & mobile apps • Use ML models & APIs • Integrate cloud/edge data	Guide projects, provide cloud credits, datasets, mentorship on ML & systems thinking	Build apps, dashboards, alerts, reports; deploy ML models; work on real datasets
3 Processing MCU, MPU, NPU, RTOS	Yes	• Program MCUs/MPUs • Use/learn RTOS basics • Explore edge AI (TinyML)	Labs for embedded systems, RTOS, edge AI; code reviews	Firmware dev, drivers, edge inference; unit testing & optimization
2 Network WiFi, BLE, LoRa, NB IoT, 4G/5G	Yes	• Use WiFi, BLE, LoRa modules • Work with NB-IoT/4G modules • Implement protocols (MQTT/CoAP)	Provide modules, SIMs, gateway kits; set up test networks	Connect devices, handle protocols, test reliability & range
1 Perception Sensors, Actuators, PCB, Power	Yes	• Interface sensors & actuators • Design simple PCBs • Handle power & battery systems	Lab instruments, PCB facility, testing equipment, safety	Prototype circuits, read sensors, drive actuators, basic PCB design & test

Should you work on all four?

- Yes, at a learning level. Go deep in one layer (your specialization) and be **competent in the others.**

What's possible in India?

- Great at student & faculty level with low-cost dev kits, open-source tools and cloud credits.
- Industry-ready projects are possible with mentorship and consistent practice.

Focus strategy

- **Years 1–2:** Build strong foundation in Layers 1 & 2
- Year 2–3: Deepen Layer 3, create real projects
- **Year 3 & beyond:** Build apps, ML, and end-to-end solutions

3-Year Learning Roadmap (Indicative)

Year 1 — Layer 1 Focus
Electronics basics, sensors, actuators, PCB, power

Year 2 — Layer 2 + 3 Focus
Communication, protocols, MCU/MPU programming, RTOS basics

Year 3 — Layer 3 + 4 Focus
Edge AI, optimization, dashboards, mobile apps, end-to-end projects

End-to-End Project Ideas

- **Smart Irrigation System**
  (Sensors → LoRa → Edge → Dashboard)
- **Health Monitoring Wearable**
  (Sensors → BLE → Mobile App → Analytics)
- **Smart Energy Monitor**
  (Meter → WiFi/NB-IoT → Cloud → Insights)

What helps most

Curiosity & Hands-on Practice · Open-source Tools · Good Mentorship · Team Projects · Hackathons & Competitions · Documentation Communication

Start small. Build often. Learn continuously. You can master the stack step by step!

How Faculties Can Enable Collaboration Across the 4 Layers

Enforce core learning for each domain, collaborate where they overlap, and make efficient use of time & knowledge

1 Layer 1: Perception

ECE Centric

Sensors, Actuators, PCB, Power

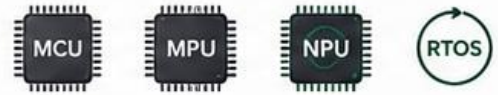
2 Layer 2: Network

ECE + CSE Overlap

WiFi, BLE, LoRa, NB-IoT, 4G/5G

3 Layer 3: Processing

ECE + CSE Overlap



MCU, MPU, NPU, RTOS

4 Layer 4: Application

CSE Centric

Dashboards, Apps, ML Models

Cross-layer Collaboration Zone (Layer 2 & 3): Communication, Edge Computing, Data Handling, Security, Performance

Faculty Collaboration Model

1. Define Core Ownership

- ECE Faculty → Own Layer 1
- CSE Faculty → Own Layer 4
- Both → Co-own Layer 2 & 3
- Clearly define learning outcomes for each layer

2. Map & Align Curriculum

- Joint curriculum map across semesters
- Identify overlap topics in Layers 2 & 3
- Avoid repetition, ensure complementarity

3. Co-teach & Share Knowledge

- Team-teaching for Layer 2 & 3 modules
- Invite faculty from other domain for focused sessions
- Regular knowledge-sharing workshops

4. Integrated Labs & Projects

- Design cross-layer lab exercises
- Mixed student teams (ECE + CSE)
- Real-world IoT use cases end-to-end
- Faculty co-mentorship

5. Efficient Use of Time

- Share resources (labs, tools, licenses, datasets)
- Use common platforms (Dev boards, cloud, simulators)
- Modular lab manuals and reusable assets

Learning Focus for Students (with Faculty Support)

Layer 1 (ECE)

- Signals, devices, circuits
- Sensors & actuators interfacing
- PCB design & prototyping
- Power management & batteries

Outcome: Build reliable hardware

Layer 2 (ECE + CSE)

- Wireless communication fundamentals
- Protocols: MQTT, CoAP, TCP/IP
- Network setup & configuration
- Security basics for IoT

Outcome: Connect devices reliably

Layer 3 (ECE + CSE)

- Embedded programming (C/C++)
- RTOS concepts
- Edge AI basics (TinyML)
- Data handling & optimization

Outcome: Process data efficiently at edge

Layer 4 (CSE)

- Web/mobile app development
- Databases & APIs
- Dashboards & visualization
- ML models & analytics

Outcome: Deliver insights & value

Example Collaborative Projects

- Smart Irrigation System
- Health Monitoring Wearable
- Smart Energy Monitor
- Environmental Monitoring Station
- Predictive Maintenance System

Implementation Path for Faculties

1. **Plan Together** — Set shared goals, define outcomes, map topics.
2. **Design Together** — Align syllabi, labs, projects & assessments.
3. **Execute Together** — Co-teach, run integrated labs & mentorship.
4. **Evaluate Together** — Use common rubrics, peer review, project evaluations.
5. **Improve Together** — Review feedback, update content, share best practices.

Benefits

- Stronger fundamentals in own domain
- Broader understanding across layers
- Better projects, placements & innovation
- Efficient use of faculty time & resources

Key Principle: Own your layer. Teach your strength. Collaborate in the overlap. Empower students to build end-to-end IoT solutions. | Together we build smarter engineers and smarter systems.

Cross-Disciplinary & Application Domains

Participating Branches & Their Roles

Branch	Primary Role
ECE	Analog design, PCB layout, RF, EMC
EEE	Power electronics, motor drives, energy
CSE	Embedded software, edge AI, cloud
Mechanical	Enclosure design, thermal, IP rating
Civil	Water systems, structural applications
Agricultural Engg	Domain expertise, deployment
Instrumentation	Sensors, calibration, traceability

Application Domains

Agriculture

- Smart irrigation & soil health
- Fertigation control
- Crop disease detection via edge AI

Water Management

- Pump automation & reservoir monitoring
- Leak detection & flow management
- Water quality sensing

Smart Cities

- Streetlight & energy management
- Waste level monitoring
- Air quality sensor networks

Industrial

- Condition monitoring & predictive maintenance
- Process control & energy metering
- Safety interlocks

Cold Chain

- Temperature & humidity tracking
- Compressor monitoring & alerting
- Last-mile cold-chain for pharma & food

Design, Testing & Certification Standards

Mandatory Design Considerations

- EMI/EMC from schematic stage
- Electrical safety & isolation
- Surge & ESD protection
- Environmental sealing (IP rating)
- Thermal management
- DFM / DFT (Design for Manufacture / Test)
- Cybersecurity for connected devices
- Calibration & traceability
- Reliability budgets (MTBF)

Progressive Testing Protocol

- Bench Tests: Function, accuracy, power draw
- Environmental: Temp cycling, humidity, vibration, IP ingress
- EMC Pre-Compliance: ESD, EFT, surge, conducted & radiated emissions
- Safety: Hipot, leakage, creepage clearance checks
- Reliability: Accelerated life testing
- Final: Formal accredited lab certification (selected designs)

Certification Pathways

- BIS (Bureau of Indian Standards)
- CE Marking (for export potential)
- FCC / IC (North America / Canada)
- IEC 61000 series (EMC)
- IEC 62368 / IEC 60950 (Safety)
- IP Rating (IEC 60529)
- Agricultural: BIS / NABL tested
- Medical: CDSCO + IEC 60601
- Funding via DST, MeitY, AICTE grants

Common Software Stack for All Processor Classes

- FreeRTOS / Zephyr RTOS
- Linux BSP (for Class C)
- TFLite Micro / ONNX Runtime for edge ML
- Modbus / MQTT / CoAP middleware
- OTA firmware update framework
- Device management & cloud dashboards
- Git-based shared firmware repository

Four-Year National Impact

Assumptions: 3,500 colleges · 200 active students/college/year · consistent execution over 4 years

35 Lakh

Students with real product-dev experience

500–1,000

Certified / certification-ready product designs

1,000s

Potential product startups spun out

35,000+

Faculty upskilled in modern product engineering

Phased 4-Year Roadmap

Phase 1 : Year 1 – Foundations

- Identified a product to develop
- System understanding starts
- Work on product definition, certification standards
- Identify Basic components, tools
- Logical thinking introduced
- C programming

Phase 2 : Years 2, 3 – Pilot & Design

- First POC in the lab with easily available components and boards
- Basics of Microcontrollers
- PCB design for EMC, Environmental
- ML fundamentals and AI
- Data collection

Phase 3 : Year 4 – Deployment

- Final Product PCB bring up
- Casing design
- Field testing in actual conditions
- ML optimization with live data
- Finalize product design documentation
- Final deployment & testing

Long-term: India builds one of the world's largest practice-based engineering ecosystems – millions of graduates with genuine product-development experience.

RE-ARCHITECTING THE FOUR-YEAR RUNWAY: ONE DEPLOYABLE PRODUCT PER TEAM

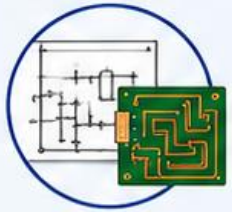
Building expertise through a progressive pipeline from fundamentals to field deployment.

STUDENT DEVELOPMENT ⟶ PIPELINE

Years 1 & 2: THE FOUNDATION

Component & Datasheet Mastery

- Read a data sheet
- Electronic design basics



Schematic & 2-Layer PCB Design

- Design schematic
- 2-layer PCB layout
- Design rule basics

Basic Analog Sensor Interfacing

- Connect simple analog sensors
- Read and analyze analog signals

Year 3: THE INTELLIGENCE

Multi-layer Hardware Bring-up

- Test and validate multi-layer hardware
- Debug with lab instruments

Bare-Metal Firmware & TinyML

- Program, embedded C/C++
- Code optimization
- TinyML development & deployment

Core System Integration

- Integrate all core components
- Validate in real-world like conditions

Year 4: THE DEPLOYMENT

Rugged IP-Rated Enclosures

- Select and design robust enclosures
- Ensure sealing, thermal and mechanical reliability

Real-world Field Stress Testing

- Environmental testing
- Long duration reliability
- Performance validation in field conditions

The Goal: A Working Physical Box

- Deployable end-product
- Ready for real-world use and demo

OUTCOMES AT THE END OF YEAR 4

Deployable Product

A fully functional, rugged, field-tested physical product.

Industry-Ready Skills

Hands-on experience across hardware, firmware, testing and system integration.

Team Collaboration

Cross-functional teamwork to build and deliver a real-world product.

Real-World Impact

Solutions that solve real problems and create measurable impact.

THE GOAL

By the end of Year 4, each team will deliver one deployable, field-tested physical product that demonstrates engineering excellence, innovation, and impact — ready for industry, incubation, or startup.




THE JOURNEY

Learn (Fundamentals) → Build (Intelligence) → Validate (Real World) → Deliver (Impact)

ENGINEERING INDIA

What I am Sharing with you

Today I am sharing:

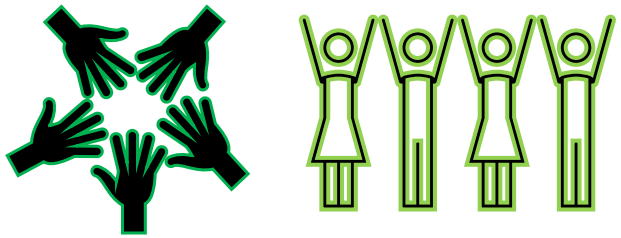
- **49 specifically identified ML & AI problems** mapped to India's most critical sectors — each problem is real, scoped, and solvable at research scale
- **A framework** for how to adopt, structure, and execute one problem as a faculty-led initiative
- **A community model** — how individual faculty effort can connect, replicate, and scale across institutions
- **Starting resources** — datasets for ML, funding pointers, tools and collaboration pathways available right now

This is not inspiration for its own sake. This is a guide map for your journey.

Outcome

The outcome possible with one action and more details around

When Students collaborate inter department at college



Students feels and learn via creating engineering solutions

- "Not everyone of us is the same, **Everyone has got a unique talent**"
- "Larger meaningful **success can come only when we come together as a TEAM**"
- "Shift from Individual heroes celebration to **win as a team for real world problem solving**"
- "My **theory grades** tells people **I am a person willing to commit my time and energy**"
- "My **hands on Engineering expertise** in action can tell everyone that **I am an able Engineer**"
- "My **Hands-on Engineering founded on theory** is **my way of service to the community & my nation**"

✓ *Student Collaboration is inspired by department faculty collaboration*
✓ *Faculties keep "My Students recognition in the nation first" and "National level college recognition" follows*

Student, Faculty, College – all WIN together

Educational Institute advantage – Enabled by Faculties

A key differentiation especially for Tier 2, Tier 3 colleges. Possibility for a national level success story.

- **The GCC Magnet:** Global Capability Centers seek "Full-Stack Hardware" talent. Your college can shift from being a "Service-Sector" feeder to become an "R&D Hub.“
- **Patent & Startup Pipeline:** Deployable hardware is the foundation for IP filing and campus-led startups.
- **Marketing Edge for the Engineering college:** “Here is a college in the region producing EMC-aware, field-deployed AI solutions and students willing to rise.“
- **Salary Transformation for student :** Moving students from 3-4 LPA (Mass Recruitment) to 8-12 LPA (Product Engineering).

Change in Student's Enthusiasm – Enabled by faculties

- **What's on the Students mind today:**
  - Which college gets most campus hiring companies. (higher number of placements, higher salary package)?
  - Will my college be able to attract campus hiring companies this year? What will make sure I get hired in the campus? What should I study besides the theory what courses I need to take?
  - Anxious students loaded with theory, hackathons, online tests what not…
- **When the Faculty and institution decides to implement 4 Year Hands-on Engineering program:**
  - **Better Clarity and confidence for students built over the years.**
  - Significant percentage of students are occupied with the mindset on what I can create?
  - Some can become entrepreneurs creating startup and other at the least have a confidence that "I have learnt a Hands-on profession ands I don't need to necessarily long for campus placement alone, I can pursue my passion".

Pros, Cons & Risk Mitigation

Pros

Educational Benefits

- Genuine hands-on experience every year
- Strong portfolios for industry hiring
- Cross-disciplinary collaboration

Industrial Benefits

- Graduates need less onboarding
- Shared reference designs cut dev cost
- Local supply-chain development

National Benefits

- Strategic self-reliance in key sectors
- Innovation & startup ecosystem growth
- Better ROI on education investment

Economic Benefits

- Commercialisation of certified designs
- Job creation in product engineering
- Long-term export potential

Cons & Challenges

Faculty Readiness

Most faculty need significant upskilling in product engineering

Funding Requirements

Lab equipment, prototyping & EMC testing are expensive

Governance Complexity

Without coordination, duplication and fragmentation occur

Evaluation Reform

Traditional exam systems don't reward hands-on work

Superficial Execution Risk

Projects may devolve into kit assembly if standards are weak

IP & Certification Cost

Formal EMC/safety testing is costly; IP ownership unclear

Key Mitigations: National steering committee · Regional shared EMC labs · Seed grants · Faculty training academies · Public benchmarking dashboards

Governance, Funding & Rollout - Suggestions

Governance Structure

National Steering Committee

- Sets standards, platform roadmap, metrics
- Includes AICTE, MeitY, DST, Industry leaders

Domain Working Groups

- Agriculture · Water · Industrial · Mobility · Healthcare
- Define application requirements & evaluation criteria

Regional Centres of Excellence

- Advanced testing, EMC pre-compliance labs
- Faculty mentoring & design reviews

Participating Colleges

- Develop modules under common standards
- Contribute to shared repository & field deployments

Funding Sources

Government

AICTE, DST, MeitY, MSME, State Government grants

Industry CSR

Sponsored problem statements, equipment, mentors

Research Grants

Mission-oriented innovation & national lab programs

Student Startups

Incubation support, TBI grants, commercialisation

Private Investment

VC / angel for certified product startups

Suggested Metrics: Working prototypes · Open reference designs · Pre-compliance passes · Certifications · Industry adoptions · Startups formed · Graduate placement outcomes

10 · Call to Action

What Every Faculty Member Can Start Tomorrow

No policy change. No government approval. Just a decision.

1 — Set the Standard

Tell your students: 'You must build one complete working product before you leave my course.'

2 — Define the Platform

Pick a sensor + embedded controller + load output combination relevant to a local problem (agriculture, water, industry).

3 — Enforce Real Testing

Require measured results, failure analysis, and documented debugging — not just a demo video.

4 — Connect with Industry

Invite one local company per semester to review student designs and provide real-world feedback.

5 — Submit to the Repository

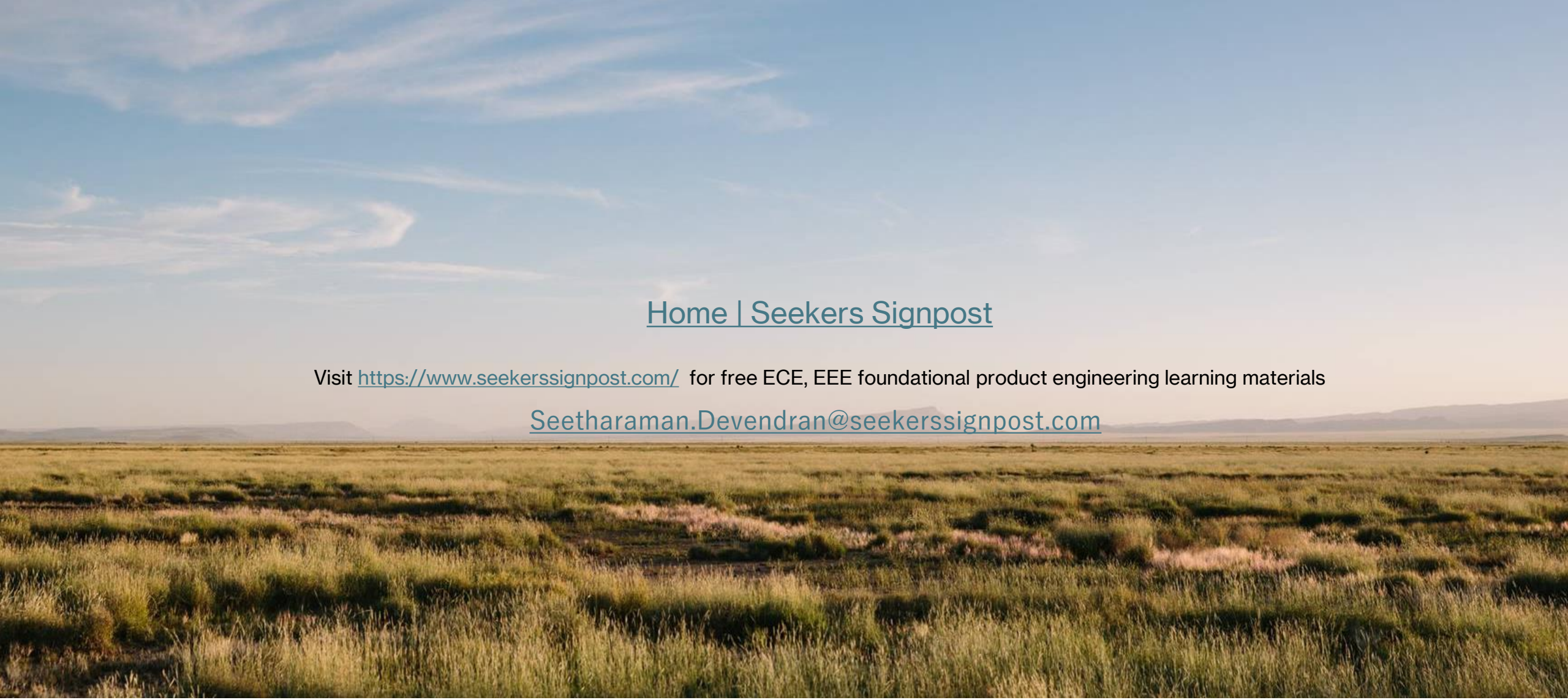
Contribute the best designs to a shared open national platform for other colleges to build upon.

6 — Push for Certification

Identify 1–2 designs each year for pre-compliance testing — pursue full certification with grant funding.

If every faculty member does this consistently — India's engineering education transforms from credential-giving to product-building.


INSTITUTE OF ENGINEERING & TECHNOLOGY
ENGINEERING EXCELLENCE FOR A BETTER TOMORROW
INNOVATE BUILD SOLVE FOR INDIA
FIELD SUCCESS!
ML based AI Solution for Agriculture
Smart Irrigation
Soil Health Monitoring
Pest & Disease Detection
Crop Yield Prediction
Deployed. Tested. Impacting Lives.
Location: College Farm
Crop: Capsicum
Model Accuracy: 94.6%
Water Saved: 32%
Yield Increase: 18%
IEPPI
AIoT NODE
SENSORS
CONNECTIVITY
AI PROCESSING
CLOUD
AI AGRICULTURE DASHBOARD
Soil Moisture
28%
Soil EC
1.2 dS/m
Temperature
29.4 °C
Humidity
62%
Irrigation Status
AUTO
NDVI Map
Pest Detection
Low Risk
ENGINEERING INNOVATION
REAL WORLD IMPACT
PROUD MOMENT.
PROUD INSTITUTION.



Home | Seekers Signpost

Visit https://www.seekerssignpost.com/ for free ECE, EEE foundational product engineering learning materials

Seetharaman.Devendran@seekerssignpost.com

Thank you

Seekers Signpost

Knowledge with Compassion for life

Q&A